**H5与Android原生互调技术文档**

**V 1.0**

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人** | **完成日期** | **版本** | **备注** |
| **李跃龙** | **2018.09.12** | **1.0** |  |
| **李跃龙** | **2018.09.13** | **1.1** | **添加Android版本适配，添加附录参考文档** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[背景 3](#_Toc524614998)

[一． 交互方式总结 3](#_Toc524614999)

[1.1交互方式 3](#_Toc524615000)

[1.1.1 Android调用JS代码 3](#_Toc524615001)

[1.1.2 JS调用Android代码 3](#_Toc524615002)

[二． 具体分析 3](#_Toc524615003)

[2.1 Android通过WebView调用JS代码 3](#_Toc524615004)

[2.1.1 方式1通过WebView的loadUrl(); 3](#_Toc524615005)

[2.1.2 方式2通过WebView的evaluateJavascript(); 5](#_Toc524615006)

[2.1.3 方法对比 6](#_Toc524615007)

[2.1.4 使用建议 6](#_Toc524615008)

[2.2 JS通过WebView调用 Android 代码 6](#_Toc524615009)

[2.2.1 通过 WebView的addJavascriptInterface（）进行对象映射 6](#_Toc524615010)

[2.2.2 通过 shouldOverrideUrlLoading ()回调拦截 url 8](#_Toc524615011)

[2.2.3 通过 WebChromeClient 的onJsAlert()、onJsConfirm()、onJsPrompt（）方法回调拦截JS对话框alert()、confirm()、prompt（） 消息 10](#_Toc524615012)

[2.2.4三种方式的对比和使用场景 13](#_Toc524615013)

[三． WebView适配 13](#_Toc524615014)

[**3.1 Android不同版本适配** 13](#_Toc524615015)

[**3.2 Android Webview适配屏幕宽度** 13](#_Toc524615016)

[四． 总结 14](#_Toc524615017)

[五． 附录 14](#_Toc524615018)

# 背景

由于项目需要，H5页面调用android原生功能以及硬件功能，如拍照，刷身份证，读写SIM卡，获取设备信息等功能，所以研究H5与原生如何互相调用，以及交互时需要注意的地方。

1. 交互方式总结

## 1.1交互方式

Android与JS通过WebView互相调用方法，实际上是：

1. Android调用JS代码
2. JS调用Android代码

二者沟通的桥梁是webview

### 1.1.1 Android调用JS代码

Android调用JS代码方式有2种

1. 通过webview的loadUrl();
2. 通过webview的evaluateJavascript();

### 1.1.2 JS调用Android代码

JS调用android代码有3种方式

1. 通过WebView的addJavascriptInterface（）进行对象映射
2. 通过 WebViewClient 的shouldOverrideUrlLoading ()方法回调拦截 url
3. 通过 WebChromeClient 的onJsAlert()、onJsConfirm()、onJsPrompt（）方法回调拦截JS对话框alert()、confirm()、prompt（） 消息
4. 具体分析

## 2.1 Android通过WebView调用JS代码

### 2.1.1 方式1通过WebView的loadUrl();

1、实例介绍：点击Android按钮，即调用WebView JS（文本名为javascript）中callJS（）

2、具体使用：

步骤1：将需要调用的JS代码以.html格式放到src/main/assets文件夹里

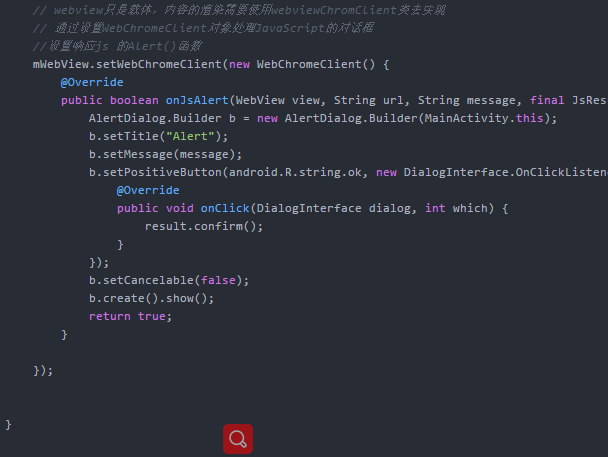
需要加载JS代码：javascript.html



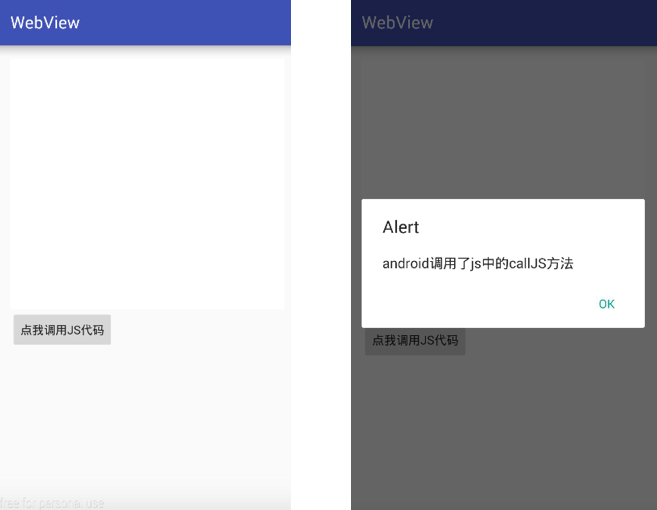
步骤2：在Android里通过WebView设置调用JS代码

Android代码：MainActivity.java





3、效果展示：



特别注意：JS代码调用一定要在 onPageFinished（） 回调之后才能调用，否则不会调用。

onPageFinished()属于WebViewClient类的方法，主要在页面加载结束时调用

### 2.1.2 方式2通过WebView的evaluateJavascript();

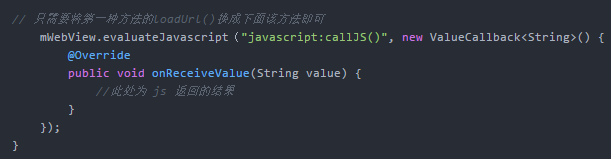
优点：该方法比第一种方法效率更高、使用更简洁。

1、因为该方法的执行不会使页面刷新，而第一种方法（loadUrl ）的执行则会。

2、Android 4.4 后才可使用

实例介绍：同2.1.1

具体使用：

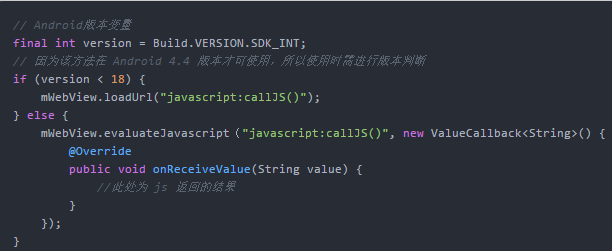


### 2.1.3 方法对比

|  |  |  |  |
| --- | --- | --- | --- |
| 调用方式 | 优点 | 缺点 | 使用场景 |
| 使用loadUrl() | 方便简洁 | 效率低，获取返回值麻烦 | 不需要获取返回值，对性能要求较低时 |
| 使用evaluateJavascript() | 效率高 | 向下兼容性差（仅android4.4以上使用） | Android4.4以上 |

### 2.1.4 使用建议

两种方法混合使用，即Android 4.4以下使用方法1，Android 4.4以上方法2



## 2.2 JS通过WebView调用 Android 代码

对于JS调用Android代码的方法有3种：

1、通过WebView的addJavascriptInterface（）进行对象映射

2、通过 WebViewClient 的shouldOverrideUrlLoading ()方法回调拦截 url

3、通过 WebChromeClient 的onJsAlert()、onJsConfirm()、onJsPrompt（）方法回调拦截JS对话框alert()、confirm()、prompt（） 消息

### 2.2.1 通过 WebView的addJavascriptInterface（）进行对象映射

步骤1：定义一个与JS对象映射关系的Android类：AndroidtoJs

AndroidtoJs.java



步骤2：将需要调用的JS代码以.html格式放到src/main/assets文件夹里

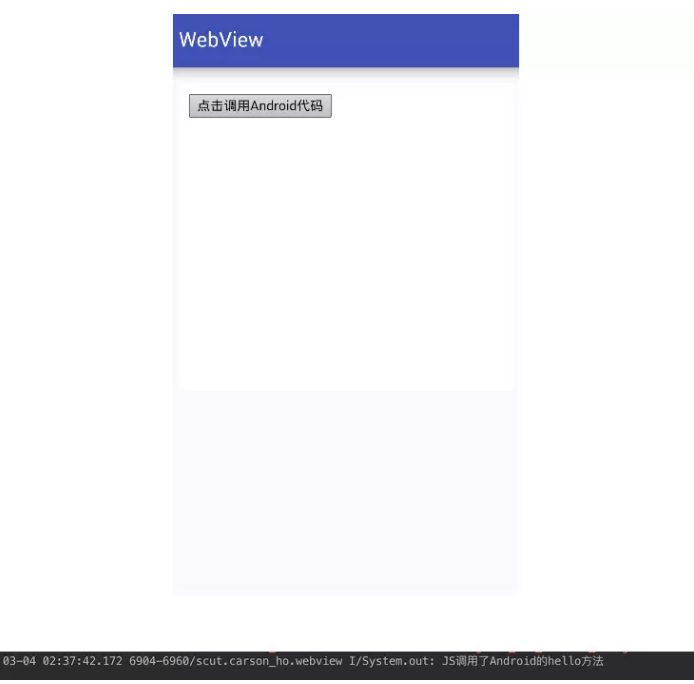
*需要加载JS代码：javascript.html*



*步骤3：在Android里通过WebView设置Android类与JS代码的映射*



效果展示：



特点

优点：使用简单，仅将Android对象和JS对象映射即可

缺点：存在严重的安全漏洞问题

### 2.2.2 通过 shouldOverrideUrlLoading ()回调拦截 url

1. 具体原理：
   1. Android通过 WebViewClient 的回调方法shouldOverrideUrlLoading ()拦截 url
   2. 解析该 url 的协议
   3. 如果检测到是预先约定好的协议，就调用相应方法
2. 具体使用

步骤1：在JS约定所需要的Url协议  
JS代码：javascript.html

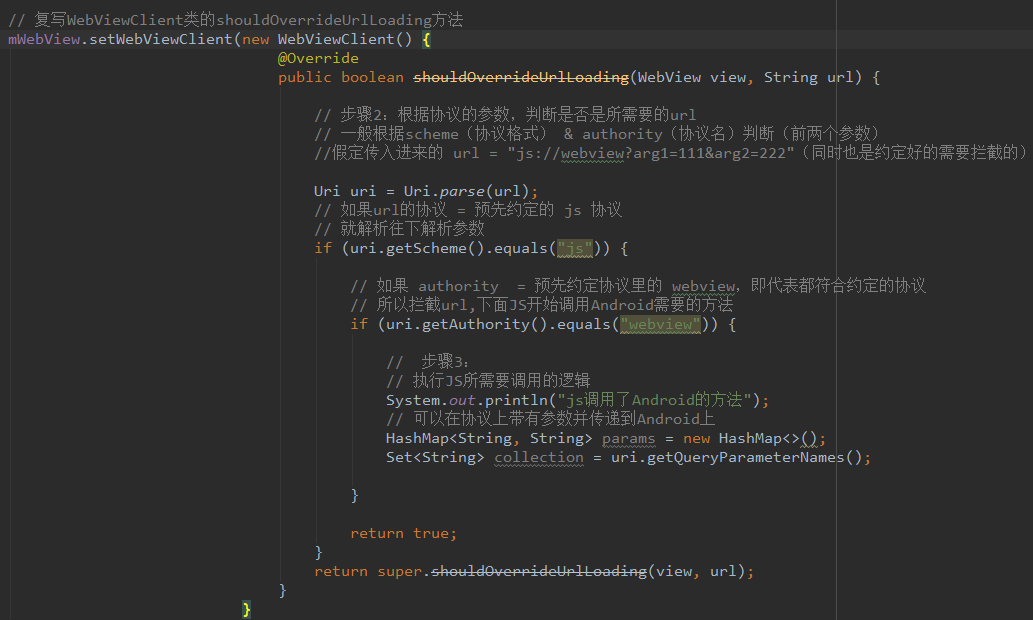
以.html格式放到src/main/assets文件夹里



当该JS通过Android的mWebView.loadUrl("file:///android\_asset/javascript.html")加载后，就会回调shouldOverrideUrlLoading （），接下来继续看步骤2：

步骤2：在Android通过WebViewClient复写shouldOverrideUrlLoading （）

MainActivity.java



效果展示：

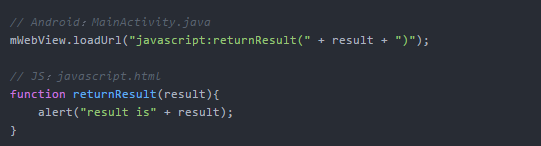


3、 特点

优点：不存在方式2.1.1中的漏洞

缺点：JS获取Android方法的返回值复杂。

如果JS想要得到Android方法的返回值，只能通过 WebView 的 loadUrl （）去执行 JS 方法把返回值传递回去，相关的代码如下：



### 2.2.3 通过 WebChromeClient 的onJsAlert()、onJsConfirm()、onJsPrompt（）方法回调拦截JS对话框alert()、confirm()、prompt（） 消息

在JS中，有三个常用的对话框方法：



1. 具体原理
   1. Android通过 WebChromeClient 的onJsAlert()、onJsConfirm()、onJsPrompt（）方法回调分别拦截JS对话框  
      （即上述三个方法），得到他们的消息内容，然后解析即可。
   2. 下面的例子将用拦截 JS的输入框（即prompt（）方法）说明 ：
   3. 常用的拦截是：拦截 JS的输入框（即prompt（）方法）
   4. 因为只有prompt（）可以返回任意类型的值，操作最全面方便、更加灵活；而alert（）对话框没有返回值；confirm（）对话框只能返回两种状态（确定 / 取消）两个值
2. 具体使用

步骤1：加载JS代码，如下：  
javascript.html

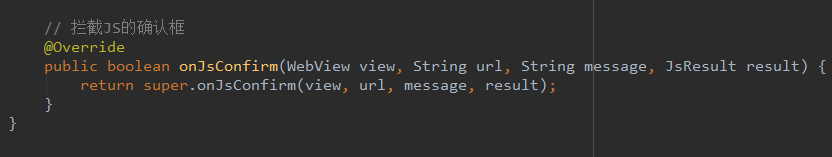
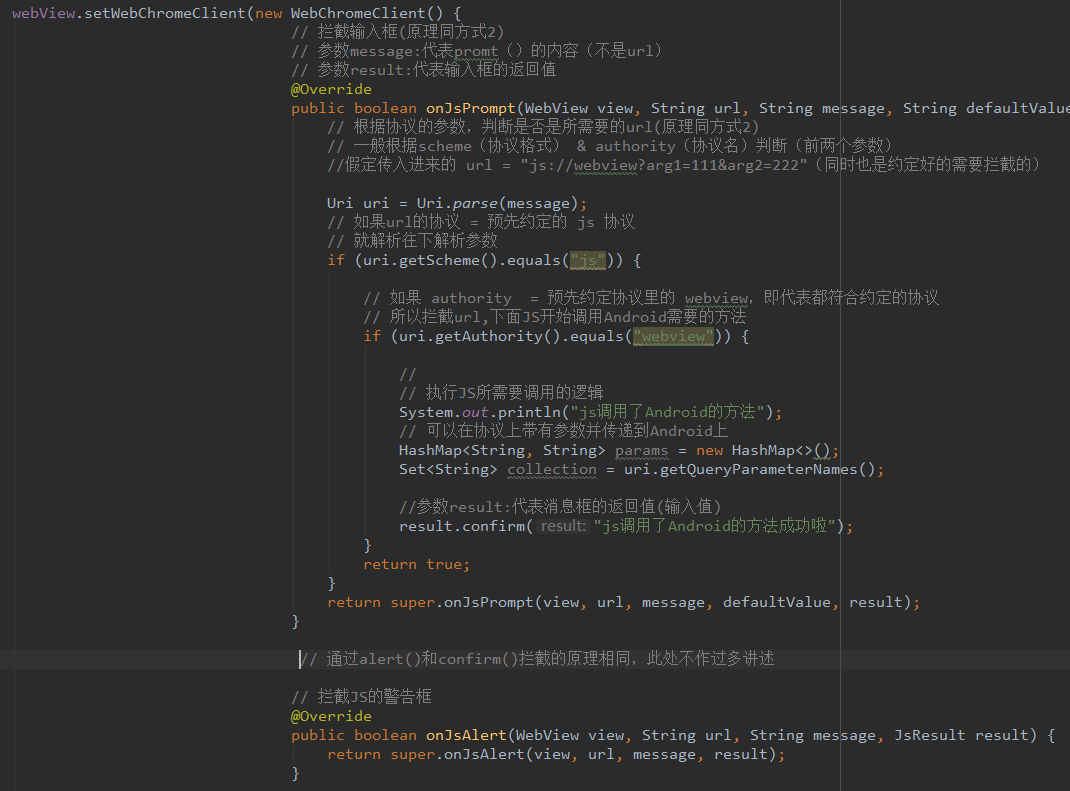
以.html格式放到src/main/assets文件夹里

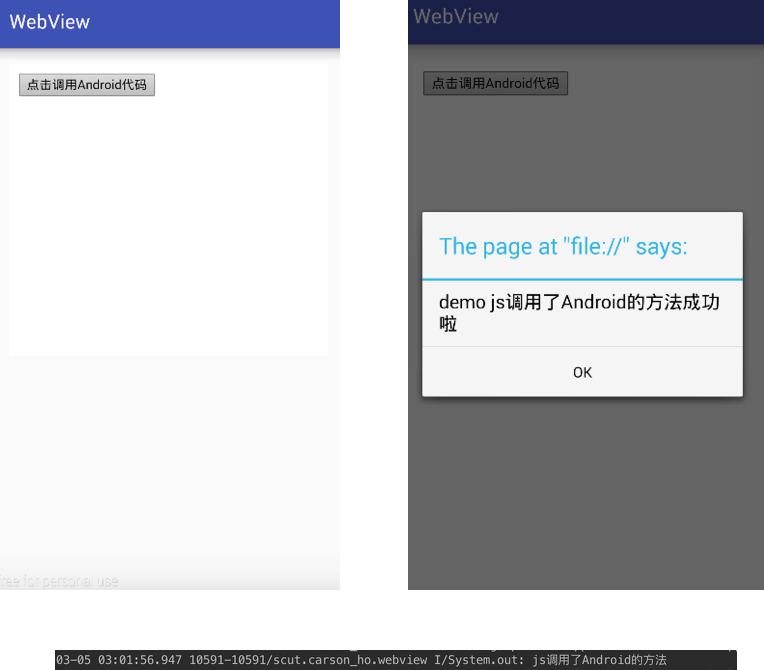


当使用mWebView.loadUrl("file:///android\_asset/javascript.html")加载了上述JS代码后，就会触发回调onJsPrompt（），具体如下：

* 1. 如果是拦截警告框（即alert()），则触发回调onJsAlert（）；
  2. 如果是拦截确认框（即confirm()），则触发回调onJsConfirm（）;

步骤2：在Android通过WebChromeClient复写onJsPrompt（）





### 2.2.4三种方式的对比和使用场景



1. WebView适配

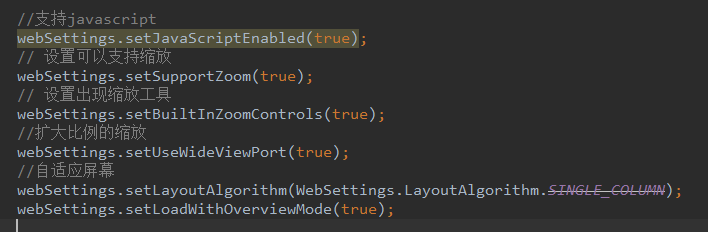
**3.1 Android不同版本适配**

WebView是Android系统提供能显示网页的系统控件，它是一个特殊的View，同时它也是一个ViewGroup可以有很多其他子View。在Android 4.4以下(不包含4.4)系统WebView底层实现是采用[WebKit内核](https://link.jianshu.com?t=http:/www.webkit.org" \t "_blank)，而在Android 4.4及其以上Google 采用了[chromium内核](https://link.jianshu.com?t=http:/www.chromium.org" \t "_blank)作为系统WebView的底层内核支持。在这一变化中Android 提供的WebView相关API并没有发生大变化，在4.4上也兼容低版本的API并且引进了少部分API。这里简单介绍下基于Chromium 的Webview和基于Webkit webview的差异，基于Chromium Webview提供更广的HTML5,CSS3,Javascript支持，在目前最新Android 系统版本上基于chromium 37，Webview提供绝大多数的HTML5特性支持。Webkit JavaScript引起采用WebCore Javascript 在Android 4.4上换成了V8能直接提升JavaScript性能。另外Chromium 支持远程调试(Chrome DevTools)。

WebKit for WebView VS Chromium for WebView性能比对

|  |  |  |  |
| --- | --- | --- | --- |
|  | Webkit for Webview | Chromium for Webview | 备注 |
| HTML5 | 278 | 434 | http://html5test.com/ |
| 远程调试 | 不支持 | 支持 | Android 4.4及以上支持 |
| 内存占用 | 小 | 大 | 相差20-30M左右 |
| WebAudio | 不支持 | 支持 | Android 5.0及以上支持 |
| WebGL | 不支持 | 支持 | Android 5.0及以上支持 |
| WebRTC | 不支持 | 支持 | Android 5.0及以上支持 |

**3.2 Android Webview适配屏幕宽度**



1. 总结

方法总结： 

1. 附录

参考文章：

1、WebView使用详解

<https://blog.csdn.net/harvic880925/article/details/51464687>

2、Android 中Webview 自适应屏幕

<https://www.jianshu.com/p/88dead61a128>

3、Android-优化不同版本系统WebView版本兼容性问题

<https://www.jianshu.com/p/e8699024d2fb>

4、Android Hybrid开发：这是一份详细 & 全面的WebView学习攻略

<https://www.jianshu.com/p/d2d4f652029d>

5、总结 Android WebView与 JS 的交互方式

<https://www.jianshu.com/p/345f4d8a5cfa>